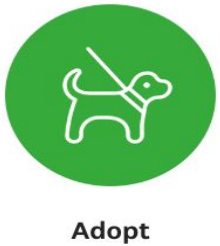
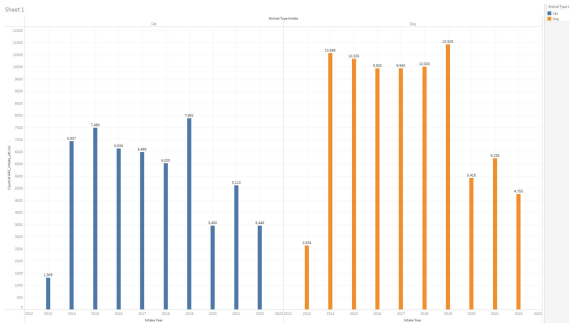
Austin Animal Center StoryBoard/Outline

Project by: Project-A-Team

Objective of Analysis: How to improve adaptability of animals in the shelter.



- Current issue: Overcrowding at Austin Animal Center.
  - Currently restricting animal intakes starting September 13th.
  - AAC Has over 700 animals.
  - https://www.kvue.com/article/life/animals/austin-animal-center-temporarily-restricts-animal-intakes/269-9fbf7f68-8682-4099-ad9c-7f84bc48c8c6?fbclid=IwAR284aVvcb8VDkzbKxpnA6VFa6OXDlg2gRj82cYLNG2IEQm-nqI1fyq3i9Y
- Graphic for yearly intakes of Cats vs Dogs throughout the years.



Overview of Analysis: What we want to solve with data storytelling.

Questions to be answered (picking our top 5 soon, waiting for our data to tell the story):

1). What month has most intakes/outcomes
2). Distribution of dog and cats.
3). Where in Austin are most intakes coming from?
4.) Average dog and cat intake in a week?
5). Average time in the shelter by age ?

Blueprint of DashBoard

1). What area of Austin is most likely to have strays? We will use a layered map in our dashboard to easily show common areas for strays.

2). What month has most intakes? Using bar graph to show month to month basis of intakes.

3). What day of the week is most likely to have adoptions? Using bar graph for days of the week to story tell what days are best for adoptions.

4). Where in Austin are most intakes coming from? Using a map to determine where most intakes are coming from.

5). Average time in shelter by age? Bar graph to show relationship between age and time.

Description of tools that will be used to create final dashboard

- We will be using tableau to create our final dashboard and will be creating our interactive elements through tableau.
- Interactive elements:
    - Filters for different years, breeds, age, maps with layers for how location is a factor, and among other interactive elements being discussed in our group.

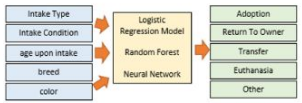
Project Workflow - Database & Machine Learning

Data Sources:
Austin Animal Center Intake Data
Austin Animal Center Outcome Data
Texas A&M GeoServices
US Cenus Bureau

PostgreSQL13
Austin Animal Center Intake Data
Austin Animal Center Outcome Data
Texas A&M GeoServices Lat and Long
US Cenus Bureau

Output Files
acc_intake_outcome.csv
acc_intake_available

Jupyter Notebook
Apply analysis and
Filter for ML



Machine Learning
Logistic Regression
Random Forest
Neural Networks

Tableua
Data Visualization and Story Telling

Data Analysis

The data analysis process began with the consideration of two datasets from Austin Animal Center.

- *Austin_Animal_Center_Intakes.csv*

Intakes represent the status of animals as they arrive at the Animal Center. All animals receive a unique Animal ID during intake. Annually over 90% of animals entering the center, are adopted, transferred to rescue or returned to their owners.

- *Austin_Animal_Center_Outcomes.csv*
The Outcomes data set reflects that Austin, TX. is the largest "No Kill" city in the country. All animals received with unique Animal IDs are safely adopted or transferred and this data represents that.

- Intake data and Outcome data contains 12 columns with 144k rows.

To analyze the Successful/Unsuccessful outcome of Adoption or relocation of pets based on data

✓ Description of preliminary data preprocessing

For preprocessing, we needed to clean up entries from each dataset and to get rid of bad or erroneous data. The columns that were entered as Strings were converted to numeric/date fields wherever possible so that the dataset is helpful and contains numeric data which will be useful for Machine Learning models. Next in the database we merged the intake and outcome datasets based on unique identifier (as described in the Database section) for further analysis.

✓ Description of preliminary feature engineering and preliminary feature selection, including their decision-making process

1. AAC_Intake_etl_step1.ipynb

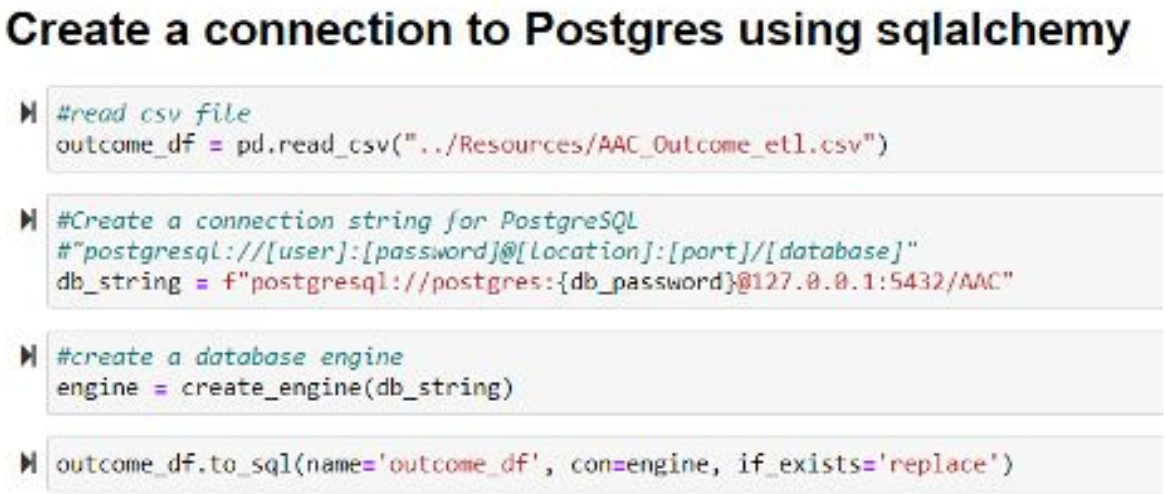
1. The "Age upon Intake" column was a string containing (days, weeks, years etc.), the column was split and then "Age Upon Intake(days)" & "Age Upon Intake(years)" was calculated.
2. Using DateTime Series Intake Month, Intake year and Intake Weekday & Intake Hour are calculated.
3. Dropping the name column.
4. Created a new column - Intake Frequency as in how many times a same animal with unique Animal ID is brought to AAC.
5. Cumulative frequency is #4 is calculated for each row based on the Intake Time and date and time in Ascending order.
6. New transformed intake_df with desired columns is put in a csv file and using postgres sql connection is loaded in the database.

2. AAC_Outcome_etl_step1.ipynb

All the above steps are repeated for the outcome data (with outgoing animal data)

3. Database Integration Description

1. Postgres SQL database was used for storing and structuring the data.
2. The files produced in the above steps were extracted using sqlalchemy to export the data into an AAC database into tables intake_df, outcome_df and zipcodes_df.



3. Once loaded into Postgres SQL tables, the files were altered to create primary key off the (“animal_id_intake”,” order_of_intake”) and (“animal_id_outcome”,” order_of_outcome”) as well as altering the date fields. The zipcode table was altered to create a primary key of “index_id” that joins to the index_id_intake.

4. Based on the new primary keys made up of a combination of Compound key from intake_df (animal_id_intake & order_of_intake) and outcome_df (animal_id_outcome & order_of_outcome), the tables are joined along with zipcode to get a combined dataset containing both data together.

5. Using case statements columns were split up and restructured for analysis such as subtypes for breeds based on the predominate identified breed, if it contained Pit Bull and date calculations.

6. The acc_intake_outcome and acc_intake_available were then exported to "acc_intake_outcome.csv" & "acc_intake_available.csv" and also connected to the machine learning script using sqlalchemy.

```
from sqlalchemy import create_engine
from config import db_password
```

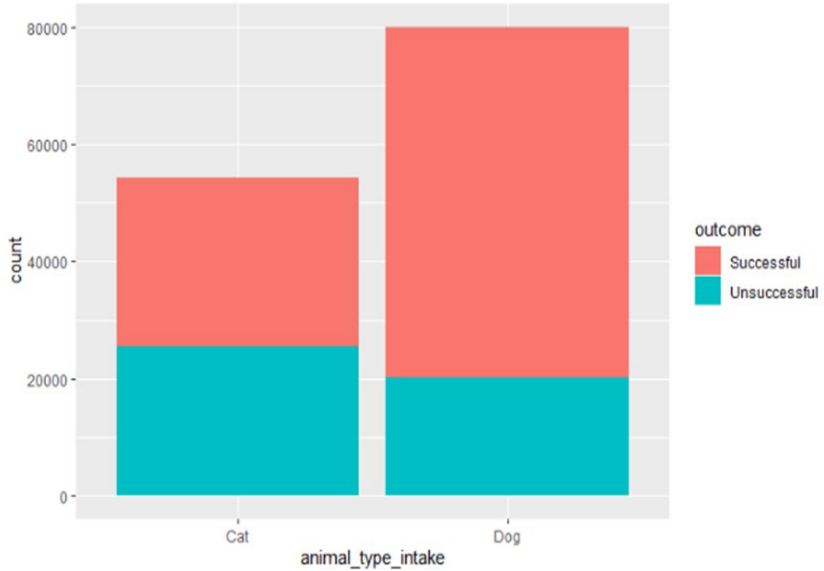
```
# Create a connection with the database in postgres
db_string = f"postgresql://postgres:{db_password}@127.0.0.1:5432/AAC"
```

```
engine = create_engine(db_string)
```

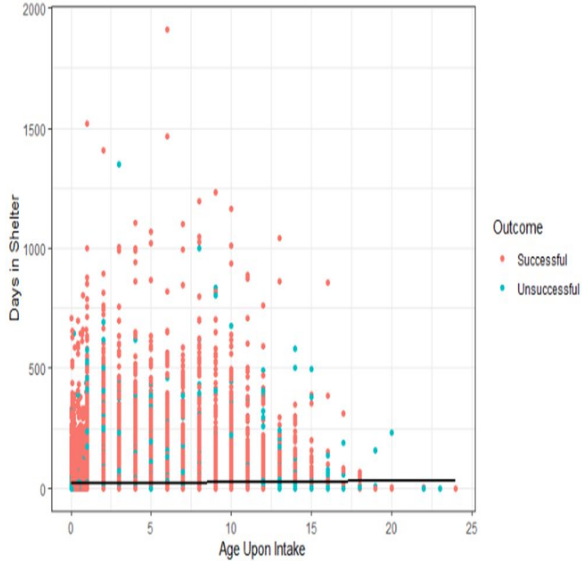
```
# read the table from the database
df = pd.read_sql_table("acc_intake_outcome",engine)
df.head()
```

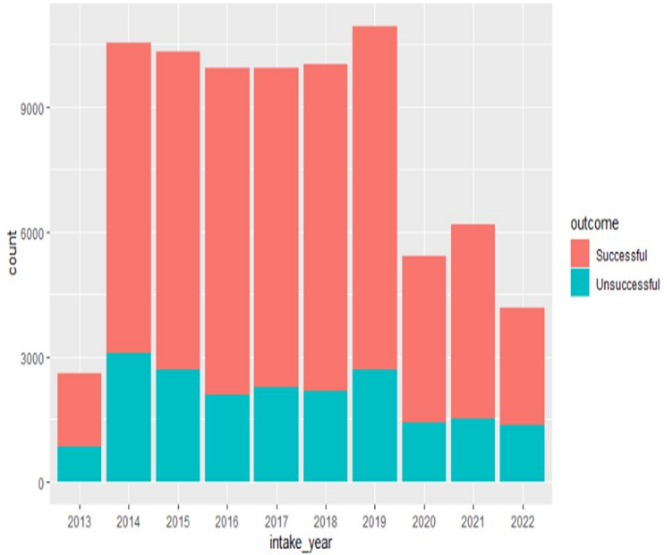
	index_id_intake	animal_id_intake	datetime_intake	monthyear_intake	found_location	intake_type	intake_condition	animal_type_intake	sex_upon_intake
0	70641	A178569	2014-03-17 09:45:00	March 2014	Austin (TX)	Public Assist	Normal	Dog	Neutered Male
					6620 Deatonhill				

Analyzing Combined Dataset - Filtering for dogs and cats only

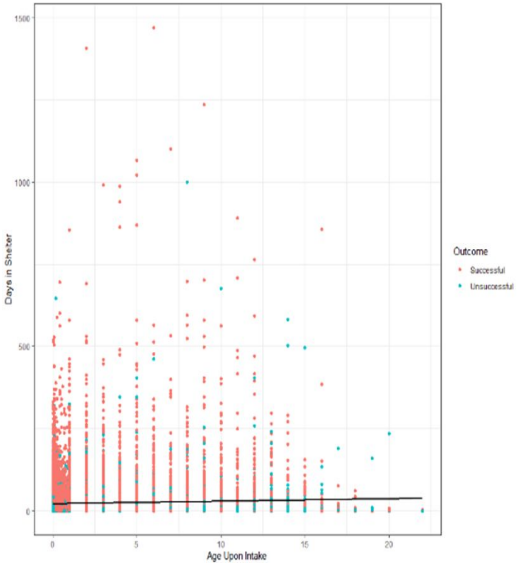


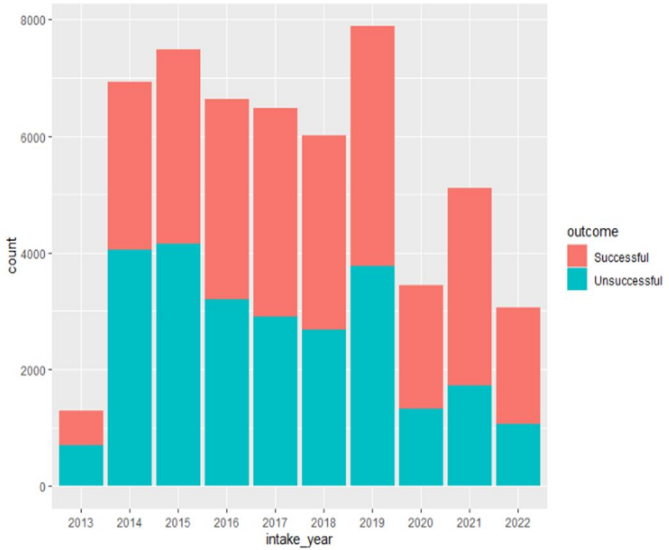
Dogs Categorized As Successful/Unsuccessful Outcome





Cats Categorized As Successful/Unsuccessful Outcome





Machine Learning Model

Code -
github - AAC_ML_model_Adopted(Success_Failure)-Dog.ipynb
github - AAC_ML_model_Adopted(Success_Failure)-Cat.ipynb &

To analyze the Successful outcome of Adoption or relocation of an Animal , we separated the datasets into 2 main category of Animals - Dogs & Cats.
To run ML model we took consideration of following categories-

1. Age
2. Breed
3. Color
4. Intake type
5. Intake condition
6. Outcome type

For Dog -

1. df_dog contains the entire data from “acc_intake_outcome.csv” for “Dog”
2. df_dog = df[df['animal_type_intake']=='Dog']
3. We checked for Unique values of all categories

We checked for Unique values of all categories

For Cat -

1. df_dog contains the entire data from “acc_intake_outcome.csv” for “Cat”
2. df_dog = df[df['animal_type_intake']==Cat]
3. We checked for Unique values of all categories

We checked for Unique values of all categories

1. Hot encoding color_intake , breed_intake, intake_type by getting the top 10 frequently occurring kinds for each of them and then merged into the main dataframe df_dog_ML. Like

```
color_intake_counts.head(10)
```

```
Black/White     9464
Brown/White     4577
White           4261
Tan/White       4204
Black           4195
Tan             3526
Brown           3208
Black/Tan       3070
Tricolor        3068
Black/Brown     2895
Name: color_intake, dtype: int64
```

```
breed_intake_counts.head(10)
```

```
Pit Bull Mix                  5332
Labrador Retriever Mix        4414
Chihuahua Shorthair Mix       3859
German Shepherd Mix           2007
Pit Bull                      1212
Australian Cattle Dog Mix     1043
Labrador Retriever             830
Chihuahua Shorthair            821
German Shepherd                696
Dachshund Mix                  687
Name: breed_intake, dtype: int64
```

```
intake_type_count
```

```
Stray                  10984
Owner Surrender         3599
Public Assist           1386
Abandoned                 85
Euthanasia Request        33
Name: intake_type, dtype: int64
```

2. intake_condition is categorized into three main categories - Normal , Aged & Other - and then they are hot encoded and merged into the main dataframe df_dog_ML

```
intake_condition_normal = ['Normal','Behavior']
intake_condition_aged = ['Aged']
intake_condtion_other = ['Injured','Sick','Nursing','Neonatal','Other','Medical','Feral','Pregnant','Med Urgent']
```

3. outcome_type is categorized into two main categories - Success & Failure - and then they are hot encoded and merged into the main dataframe df_dog_ML.


```
intake_condition_normal = ['Normal','Behavior']
intake_condition_aged = ['Aged']
intake_condtion_other = ['Injured','Sick','Nursing','Neonatal','Other','Medical','Feral','Pregnant','Med Urgent']
```

3. outcome_type is categorized into two main categories - Success & Failure - and then they are hot encoded and merged into the main dataframe df_dog_ML.

```
: # # Determine which values to replace
  # replace_intake_condition = list(intake_type[intake_type < 465].index)

  other_outcome_type_list = ['Transfer','Euthanasia','Died','Disposal','Missing']
  success_outcome_list = ['Adoption','Return to Owner','Rto-Adopt']


  # # Replace in DataFrame
  for outcome in other_outcome_type_list:
      df_dog_ML.outcome_type = df_dog_ML.outcome_type.replace(outcome,"Failure")

  for outcome in success_outcome_list:
      df_dog_ML.outcome_type = df_dog_ML.outcome_type.replace(outcome,"Success")


  # # Check to make sure binning was successful
  df_dog_ML.outcome_type.value_counts()
```

Then we used the function get_dummies on df_dog_ML["outcome_type"] and then merging into the dataframe df_dog_ML.

✓ Description of how data was split into training and testing sets

The data is split into X and y. Where below are the features -
X = The hot encoded values of the following features

- ‘age_upon_intake(days)’
- ’age_upon_outcome(days)’
- ’days_in_shelter’
- ’intake_condition’
- ‘color_intake’
- ‘breed_intake’
- ’intake_type’

Y = ‘outcome_type’ encoded for Success and Failure.

```
# Seperate the features X from the target Y
y = df_dog_ML.Success
columns=["Success","Failure"]
X = df_dog_ML.drop(columns=columns)


# Split training/test datasets
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42, stratify=y)
```

✓ Explanation of model choice, including limitations and benefits.

We used Logistic Regression and Random Forest Classification models to analyze the data.

Logistic Regression - is performed when we are expecting a Binary Outcome - Here we are running the ML model to determine if Dog / Cat will have success or Failure as outcome for given categories or features in consideration.
Random Forest Classification - This model produces good predictions, and is capable to handle large datasets efficiently. This model helps in producing higher level of accuracy. Below is the Confusion Matrix for Dogs and Cats.

ML results For Dogs

Confusion matrix -

```
# Displaying results
print("Confusion Matrix for Dogs")
display(cm_df)
print(f"Accuracy Score : {acc_score}")
print("Classification Report")
print(classification_report(y_test, predictions))
```

Confusion Matrix for Dogs

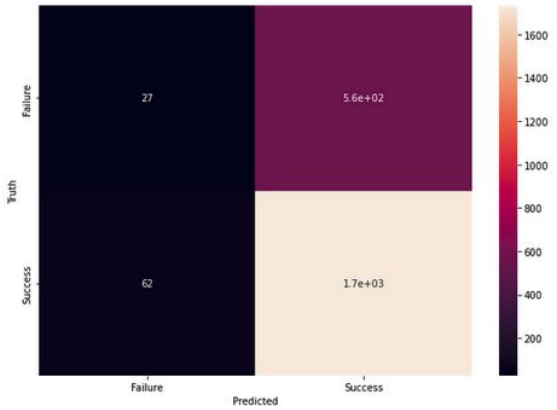
	Failure	Success
Failure	27	565
Success	62	1732

```
Accuracy Score : 0.7372170997485331
Classification Report
              precision    recall  f1-score   support

           0       0.30      0.05      0.08       592
           1       0.75      0.97      0.85      1794

    accuracy                           0.74      2386
   macro avg       0.53      0.51      0.46      2386
weighted avg       0.64      0.74      0.66      2386
```

Plotting of confusion matrix



ML results For Cats

Confusion Matrix

```
# Displaying results
print("Confusion Matrix for Cats.")
display(cm_df)
print(f"Accuracy Score : {acc_score}")
print("Classification Report")
print(classification_report(y_test, predictions))
```

Confusion Matrix for Cats.

	Failure	Success
Failure	79	67
Success	82	76

```
Accuracy Score : 0.5098684210526315
Classification Report
              precision    recall  f1-score   support

           0       0.49      0.54      0.51       146
           1       0.53      0.48      0.50       158

    accuracy                           0.51       304
   macro avg       0.51      0.51      0.51       304
weighted avg       0.51      0.51      0.51       304
```

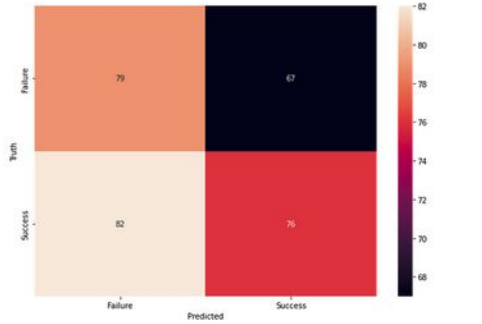
Plotting of confusion matrix

```
#Plotting a confustion matrix using matplotlib

%matplotlib inline
import matplotlib.pyplot as plt

import seaborn as sn
plt.figure(figsize=(10,7))
sn.heatmap(cm_df,annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

```
Text(69.0, 0.5, 'Truth')
```



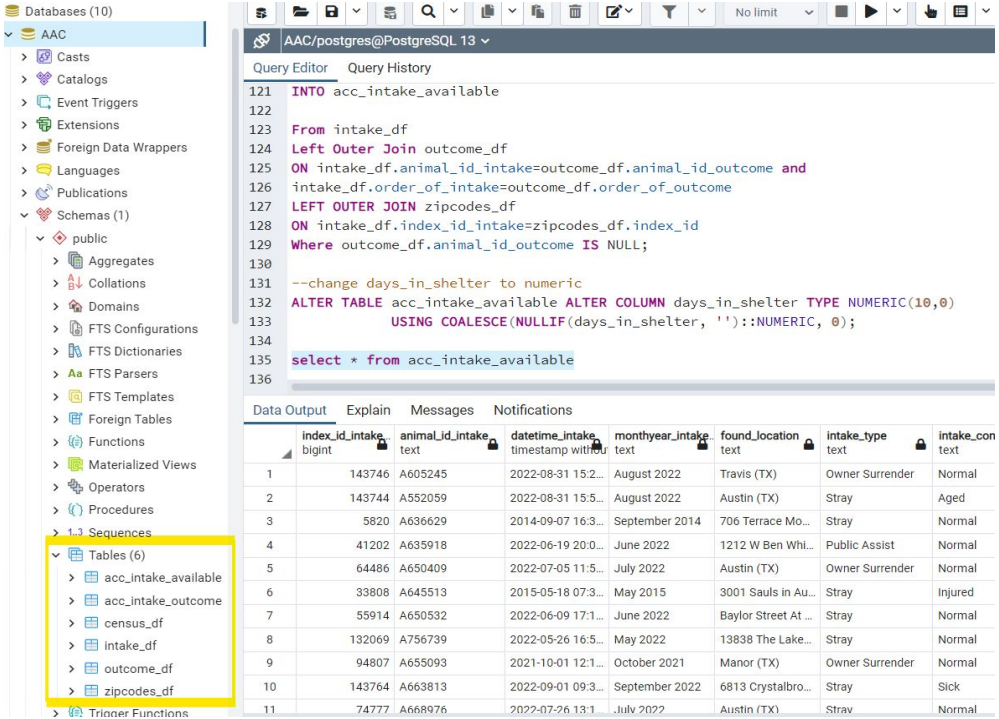
Database

Database Integration

For this project, we utilized PostgresSQL and fully integrated the database into our project.

- Database stores static data for use during the project
  - *Database stores multiple data tables used for compiling the final dataset.*
- Database interfaces with the project in some format (e.g., scraping updates the database)
  - *Database interfaces with the project using Jupyter Notebook Pandas and sqlalchemy to export the dfs into the Postgres AAC database and create the tables and to import the final sql table back to Jupyter Notebooks to use for the machine learning.*
- Includes at least two tables (or collections, if using MongoDB)
  - *The AAC database includes three tables used for the final dataset.*
- Includes at least one join using the database language (not including any joins in Pandas)
  - *SQL is used to join the three tables together and perform data manipulation on number, character and date columns.*
- Includes at least one connection string (using SQLAlchemy or PyMongo)
  - *Three connection strings using SQLAlchemy export the data into the Postgres database and two connection strings import the final dataset into the Jupyter Notebook machine learning scripts.*

Postgres Database Stores Data Tables



Database Interfaces With AAC_ML_model_Adopted(Success_Failure) - Dog/Cat

In [2]:
```
import pandas as pd
import sklearn as skl
from sklearn.preprocessing import OneHotEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
```

In [3]:
```
from sqlalchemy import create_engine
from config import db_password
```

In [4]:
```
# Create a connection with the database in postgres
db_string = f"postgresql://postgres:{db_password}@127.0.0.1:5432/AAC"
```

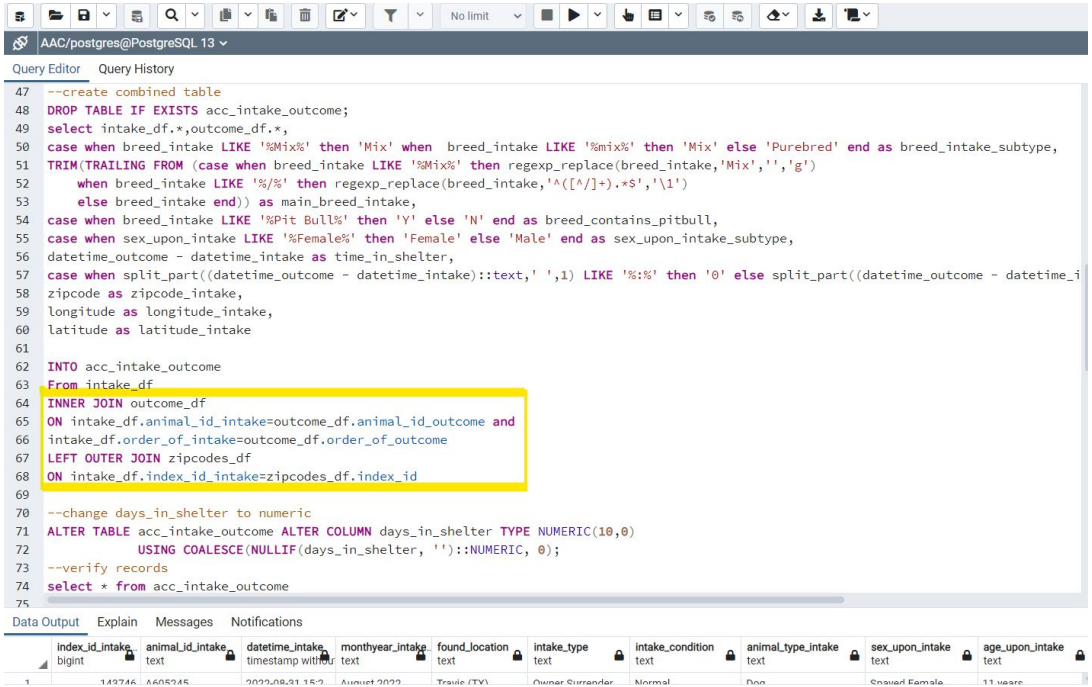
In [5]:
```
engine = create_engine(db_string)
```

In [6]:
```
# read the table from the database
df = pd.read_sql_table("acc_intake_outcome",engine)
df.head()
```

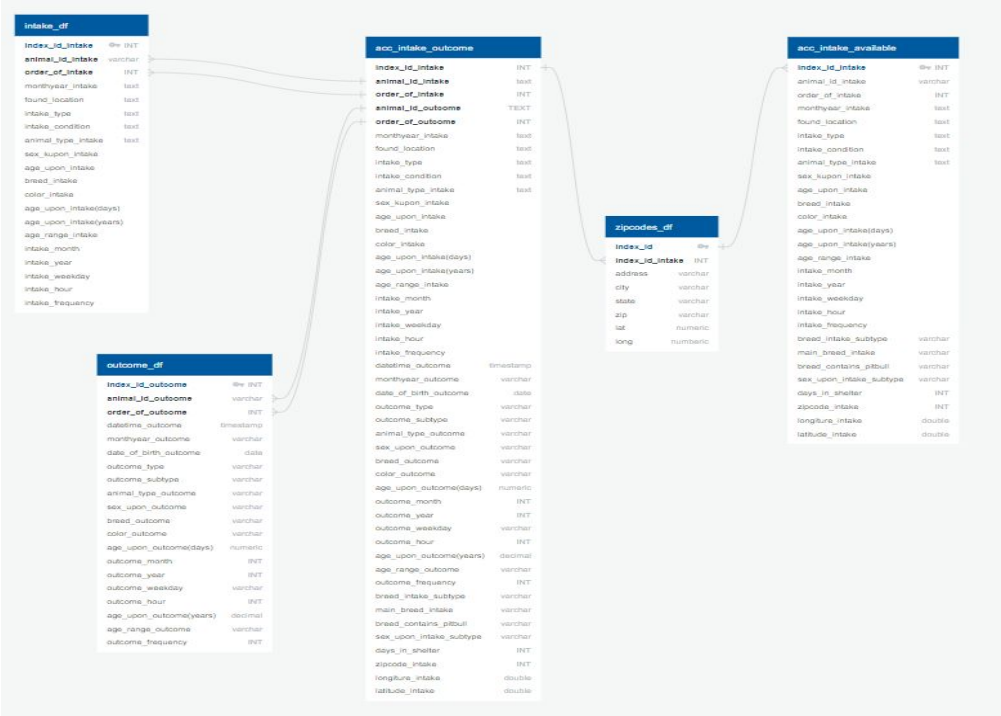
Out[6]:

	index_id_intake	animal_id_intake	datetime_intake	monthyear_intake	found_location	intake_type	intake_condition	animal_type_intake	sex_upon_intake
0	70641	A178569	2014-03-17 09:45:00	March 2014	Austin (TX)	Public Assist	Normal	Dog	Neutered Male
1	944	A287017	2015-08-16 12:19:00	August 2015	6620 Deatonhill Dr in Austin (TX)	Stray	Normal	Dog	Spayed Female
2	69127	A293383	2018-03-18 18:17:00	March 2018	6005 Walnut Hills in Austin (TX)	Stray	Sick	Cat	Neutered Male

Database Includes Joins. Refer to full SQL @ /Database/aac.sql



ERD


intake_df
index_id_intake PK INT
animal_id_intake varchar
order_of_intake INT
monthyear_intake text
found_location text
intake_type text
intake_condition text
animal_type_intake text
sex_kupon_intake
age_upon_intake
breed_intake
color_intake
age_upon_intake(days)
age_upon_intake(years)
age_range_intake
intake_month
intake_year
intake_weekday
intake_hour
intake_frequency
outcome_df
index_id_outcome PK INT
animal_id_outcome varchar
order_of_outcome INT
datetime_outcome timestamp
monthyear_outcome varchar
date_of_birth_outcome date
outcome_type varchar
outcome_subtype varchar
animal_type_outcome varchar
sex_upon_outcome varchar
breed_outcome varchar
color_outcome varchar
age_upon_outcome(days) numeric
outcome_month INT
outcome_year INT
outcome_weekday varchar
outcome_hour INT
age_upon_outcome(years) decimal
age_range_outcome varchar
outcome_frequency INT
acc_intake_outcome
index_id_intake INT
animal_id_intake text
order_of_intake INT
animal_id_outcome TEXT
order_of_outcome INT
monthyear_intake text
found_location text
intake_type text
intake_condition text
animal_type_intake text
sex_kupon_intake
age_upon_intake
breed_intake
color_intake
age_upon_intake(days)
age_upon_intake(years)
age_range_intake
intake_month
intake_year
intake_weekday
intake_hour
intake_frequency
datetime_outcome timestamp
monthyear_outcome varchar
date_of_birth_outcome date
outcome_type varchar
outcome_subtype varchar
animal_type_outcome varchar
sex_upon_outcome varchar
breed_outcome varchar
color_outcome varchar
age_upon_outcome(days) numeric
outcome_month INT
outcome_year INT
outcome_weekday varchar
outcome_hour INT
age_upon_outcome(years) decimal
age_range_outcome varchar
outcome_frequency INT
bread_intake_subtype varchar
main_bread_intake varchar
bread_contains_pitbull varchar
sex_upon_intake_subtype varchar
days_in_shelter INT
zipcode_intake INT
longitude_intake double
latitude_intake double
zipcodes_df
index_id PK
index_id_intake INT
address varchar
city varchar
state varchar
zip varchar
lat numeric
long numeric
acc_intake_available
index_id_intake PK INT
animal_id_intake varchar
order_of_intake INT
monthyear_intake text
found_location text
intake_type text
intake_condition text
animal_type_intake text
sex_kupon_intake
age_upon_intake
breed_intake
color_intake
age_upon_intake(days)
age_upon_intake(years)
age_range_intake
intake_month
intake_year
intake_weekday
intake_hour
intake_frequency
bread_intake_subtype varchar
main_bread_intake varchar
bread_contains_pitbull varchar
sex_upon_intake_subtype varchar
days_in_shelter INT
zipcode_intake INT
longitude_intake double
latitude_intake double

Conclusion/Proposals

1). Drawing conclusions from our questions we solve with our data analysis/storytelling.

2). Propose ideas on how AAC could improve adoptability by targeting animals that are most adoptable to get them out of the shelter faster.

3). Propose new adoption programs that can improve adoption rates of older animals, lower cost upfront, extra supplies, and more.

4). Show pers how many pets are available in a local shelter in Austin.

Segment 1 Deliverable

Austin Animal Center Adoption Project

Presented by: A-Team

Purpose:

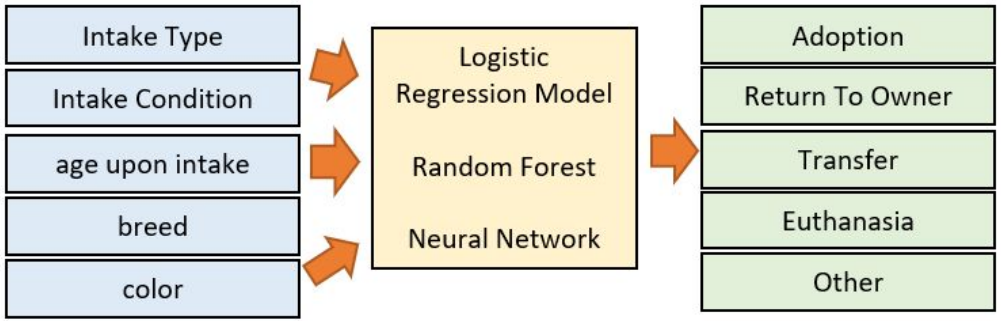
The Austin Animal Center is located in Austin, Texas, the "No Kill" city in the country. The center reports that 90% of animals are either adopted, returned to their owner, or transferred to a rescue. Our interest in this topic is to determine what outcome would be expected for each breed, particularly for breeds considered “violent” or aggressive, and compare if younger or older pets are more likely to find a home. Another interest we had was looking at the geographical area of the City of Austin to determine if there is an area within the city that has a higher number of stray population.

Questions to Answer: (Still being discussed)

1). What month has most intakes/outcomes
2). Distribution of dog and cats.
3). Where in Austin are most intakes coming from?
4.) Average dog and cat intake in a week?
5). Average time in the shelter?
6). Average time in shelter by age?
7). What type of animal is most likely to be returned to the owner?
8). What area of Austin is most likely to have strays?
9.) What day of the week is most likely to have adoptions?
10.) Does breed play a role in successful outcome?

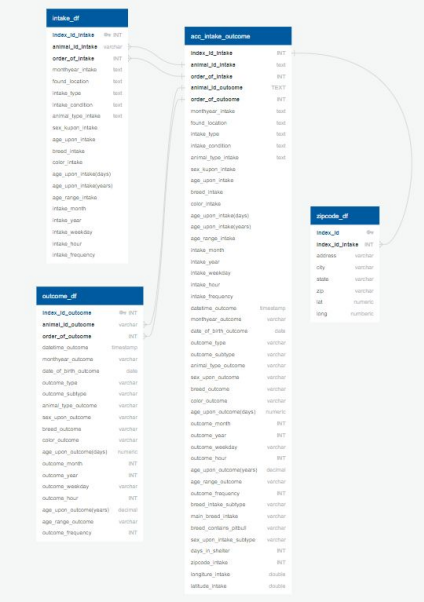
Workflow of Project: Machine Learning Model

The following is a provisional machine learning model which was used during the planning



Workflow of Project: Database

Database ERD



Resources: Data

City of Austin Data Center
(https://data.austintexas.gov/browse?City-of-Austin_Department-=Animal+Services)
- The city of Austin's online repository of statistical data. From this data source we downloaded Animal Center Intake and Outcome data from Oct, 1st 2013 to present. All animals receive a unique Animal ID during intake.

Intake Data
(https://data.austintexas.gov/Health-and-Community-Services/Austin-Animal-Center-Intakes/wter-evkm/data)
- Intake represents the status of animals as they enter the Animal Center

Outcome Data
(https://data.austintexas.gov/Health-and-Community-Services/Austin-Animal-Center-Outcomes/9t4d-g238/data)
- Outcome represents the status of animals as they leave the Animal Center